6.1.1.2 Frame

A frame consists of three rectangular matrices of integers: a luminance matrix (Y), and two chrominance matrices (Cb and Cr).

The relationship between these Y, Cb and Cr components and the primary (analogue) Red, Green and Blue Signals (E'_R, E'_G and E'_B), the chromaticity of these primaries and the transfer characteristics of the source frame may be specified in the bitstream (or specified by some other means). This information does not affect the decoding process.

6.1.1.3 Field

A field consists of every other line of samples in the three rectangular matrices of integers representing a frame.

A frame is the union of a top field and a bottom field. The top field is the field that contains the top-most line of each of the three matrices. The bottom field is the other one.

6.1.1.4 Picture

A reconstructed picture is obtained by decoding a coded picture, i.e. a picture header, the optional extensions immediately following it, and the picture data. A coded picture may be a frame picture or a field picture. A reconstructed picture is either a reconstructed frame (when decoding a frame picture), or one field of a reconstructed frame (when decoding a field picture).

6.1.1.4.1 Field pictures

If field pictures are used then they shall occur in pairs (one top field followed by one bottom field, or one bottom field followed by one top field) and together constitute a coded frame. The two field pictures that comprise a coded frame shall be encoded in the bitstream in the order in which they shall occur at the output of the decoding process.

When the first picture of the coded frame is a P-field picture, then the second picture of the coded frame shall also be a P- field picture. Similarly when the first picture of the coded frame is a B-field picture the second picture of the coded frame shall also be a B-field picture.

When the first picture of the coded frame is a I-field picture, then the second picture of the frame shall be either an I-field picture or a P-field picture. If the second picture is a P-field picture then certain restrictions apply, see 7.6.3.5.

6.1.1.4.2 Frame pictures

When coding interlaced sequences using frame pictures, the two fields of the frame shall be interleaved with one another and then the entire frame is coded as a single frame-picture.

6.1.1.5 Picture types

There are three types of pictures that use different coding methods.

An **Intra-coded (I) picture** is coded using information only from itself.

A **Predictive-coded (P) picture** is a picture which is coded using motion compensated prediction from a past reference frame or past reference field.

A **Bidirectionally predictive-coded (B) picture** is a picture which is coded using motion compensated prediction from a past and/or future reference frame(s).

6.1.1.6 Sequence header

A video sequence header commences with a sequence_header_code and is followed by a series of data elements. In this specification sequence_header() shall be followed by sequence_extension() which

6.2.2.3 Sequence extension

sequence_extension() {	No. of bits	Mnemonic
extension_start_code	32	bslbf
extension_start_code_identifier	4	uimsbf
profile_and_level_indication	8	uimsbf
progressive_sequence	1	uimsbf
chroma_format	2	uimsbf
horizontal_size_extension	2	uimsbf
vertical_size_extension	2	uimsbf
bit_rate_extension	12	uimsbf
marker_bit	1	bslbf
vbv_buffer_size_extension	8	uimsbf
low_delay	1	uimsbf
frame_rate_extension_n	2	uimsbf
frame_rate_extension_d	5	uimsbf
next_start_code()		
}		

6.2.2.4 Sequence display extension

sequence_display_extension() {	No. of bits	Mnemonic
extension_start_code_identifier	4	uimsbf
video_format	3	uimsbf
colour_description	1	uimsbf
if (colour_description) {		
colour_primaries	8	uimsbf
transfer_characteristics	8	uimsbf
matrix_coefficients	8	uimsbf
}		
display_horizontal_size	14	uimsbf
marker_bit	1	bslbf
display_vertical_size	14	uimsbf
next_start_code()		
}		

display_horizontal_size and display_vertical_size do not affect the decoding process but may be used by the display process that is not standardised in this specification.

6.3.7 Sequence scalable extension

It is a syntactic restriction that if a sequence_scalable_extension() is present in the bitstream following a given sequence_extension() then sequence_scalable_extension() shall follow every other occurrence of sequence_extension(). Thus a bitstream is either scalable or it is not scalable. It is not possible to mix scalable and non-scalable coding within a sequence.

scalable_mode -- The scalable_mode indicates the type of scalability used in the video sequence. If no sequence_scalable_extension() is present in the bitstream then no scalability is used for that sequence. scalable_mode also indicates the macroblock_type tables to be used. However in the case of spatial scalability if no picture_spatial_scalable_extension() is present for a given picture then that picture shall be decoded in a non-scalable manner (i.e. as if sequence_scalable_extension() had not been present).

Table 6-10. Definition of scalable_mode

scalable_mode	Meaning	picture_spatial_scalable-_extension()	macroblock_type tables
sequence_scalable_extension() not present			B-2, B-3 and B-4
00	data partitioning	--	B-2, B-3 and B-4
01	spatial scalability	present	B-5, B-6 and B-7
		not present	B-2, B-3 and B-4
10	SNR scalability		B-8
11	temporal scalability		B-2, B-3 and B-4

layer_id -- This is an integer which identifies the layers in a scalable hierarchy. The base layer always has layer_id = 0. However the base layer of a scalable hierarchy does not carry a sequence_scalable_extension() and hence layer_id, except in the case of data partitioning. Each successive layer has a layer_id which is one greater than the layer for which it is an enhancement.

In the case of data partitioning layer_id shall be zero for partition zero and layer_id shall be one for partition one.

lower_layer_prediction_horizontal_size -- this is a 14-bit integer indicating the horizontal size of the lower layer frame which is used for prediction. This shall contain the value contained in horizontal_size (horizontal_size_value and horizontal_size_extension) in the lower layer bitstream.

lower_layer_prediction_vertical_size -- this is a 14-bit integer indicating the vertical size of the lower layer frame which is used for prediction. This shall contain the value contained in vertical_size (vertical_size_value and vertical_size_extension) in the lower layer bitstream.

horizontal_subsampling_factor_m -- This affects the spatial scalable upsampling process, as defined in 7.7.2. The value zero is forbidden.

horizontal_subsampling_factor_n -- This affects the spatial scalable upsampling process, as defined in 7.7.2. The value zero is forbidden.

vertical_subsampling_factor_m -- This affects the spatial scalable upsampling process, as defined in 7.7.2. The value zero is forbidden.

vertical_subsampling_factor_n -- This affects the spatial scalable upsampling process, as defined in 7.7.2. The value zero is forbidden.

picture_mux_enable -- If set to 1, picture_mux_order and picture_mux_factor are used for remultiplexing prior to display.